Making Real Estate Walkthrough Videos Interactive

Mathijs Lens[a], Floris De Feyter[b] and Toon Goedemé[c]
EAVISE-PSI, KU Leuven, Campus De Nayer, Sint-Katelijne-Waver, Belgium
{*mathijs.lens, toon.goedeme*}*@kuleuven.be*

Keywords: Video Segmentation, Transformer, TCN.

Abstract: This paper presents an automated system designed to streamline the creation of interactive real estate video tours. These virtual walkthrough tours allow potential buyers to explore properties by skipping or focusing on rooms of interest, enhancing the decision-making process. However, the current manual method for producing these tours is costly and time-consuming. We propose a system that automates key aspects of the walkthrough video creation process, including the identification of room transitions and room label extraction. Our proposed system utilizes transformer-based video segmentation, addressing challenges such as the lack of clear visual boundaries between open-plan rooms and the difficulty of classifying rooms in unfurnished properties. We demonstrate in an ablation study that the combined usage of ResNet frame embeddings, and a transformer-based temporal postprocessing that uses a separately trained doorway detection network as extra input yields the best results for room segmentation and classification. This method improves the edit score by +35% compared to frame-by-frame classification. All experiments are performed on a large real-life dataset of 839 walkthrough videos.

1 INTRODUCTION

The process of searching for new property has increasingly moved to digital platforms, with 76% of people using their phones or tablets to explore potential properties and more and more people using social media for their property search (Lautz et al., 2014). This shift has prompted real estate sellers to adopt digital representations more suited to mobile users. Among these representations, interactive video tours stand out by offering a comprehensive understanding of a property's structure and appearance compared to separate images. These tours provide potential buyers with an immersive experience, allowing them to get a better feel of the property without physically being there. Interactive video tours enable users to virtually walk through a property with the option to skip to rooms of interest. This functionality allows viewers to bypass less interesting areas, such as hallways, and focus on more important spaces like bathrooms and bedrooms, effectively speeding up their decision-making process. However, creating these interactive tours is a labor-intensive task that involves two video editing steps:

- Cutting the video into smaller clips per room
- Labelling each clip with the correct room name

Despite the clear advantages of interactive video tours, the manual process of creating them is time-consuming and subject to human bias. These steps require significant effort and expertise in video editing, which can be a barrier for many real estate professionals.

In this work, we propose a system to automatically process a walkthrough video into an interactive video for real estate interactive tours. Our pipeline is designed to: (1) identify transitions between rooms, and (2) extract room labels for each video frame. Examples of such interactive video tours can be seen at https://youtu.be/XQqFN4KsX_A and https://youtu.be/bZBrMz2eGtM.

The input to our pipeline is a video which is manually captured with a mobile phone while walking through the property for sale. This is typically done by the real estate agent, who is given specific instructions. Every room in the house needs to be filmed, including the street scene in front of the property, the garden, terrace, etc.

Our automated approach aims to reduce the time and effort required to produce high-quality interactive video tours, thereby enhancing the efficiency of real estate marketing. By leveraging advanced video

[a] https://orcid.org/0009-0005-4798-3555
[b] https://orcid.org/0000-0003-2690-0181
[c] https://orcid.org/0000-0002-7477-8961

Lens, M., De Feyter, F. and Goedemé, T.
Making Real Estate Walkthrough Videos Interactive.
DOI: 10.5220/0013126200003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 3: VISAPP*, pages 319-326
ISBN: 978-989-758-728-3; ISSN: 2184-4321

segmentation and processing techniques, our system promises to deliver consistent and accurate results, making the creation of interactive video tours more accessible and scalable. Instead of the 25 minutes the fully manual process typically takes, our pipeline reduces the manual effort to a quick and simple quality check of the automatically produced outcome.

The two tasks—room transition detection and room classification—essentially boil down to a temporal video segmentation and classification problem. This is akin to the video action segmentation problem (Lea et al., 2016; Lea et al., 2017a; Miech et al., 2020), with a high focus on exact transition placement. Here, we divide long videos into their respective room segments, the "actions". Doing this step manually often requires multiple inspections of the same video to get the labels and the transitions right, making the process not scalable and expensive. An automatic approach would be highly beneficial.

At first sight, a simple frame-by-frame room type classifier would suffice to solve this problem. However, the problem is much more difficult as rooms are not always clearly delineated from each other by doors. For instance, in open-plan kitchens, there is no clear point where the kitchen ends and the dining room or living room starts. Moreover, from a single frame view, the room type is indiscernible in many cases because of too few room-specific items in view. Often, houses are sold in unfurnished state, which makes the room type classification even harder.

In this paper, we present a multi-cue transformer-based approach to solve this room video segmentation problem. We will train and test our approach on a large dataset of real-life real estate walkthrough videos, encompassing various types of houses in both furnished and unfurnished states.

2 RELATED WORK

The two tasks central to our pipeline—room transition detection and room classification—are fundamentally temporal video segmentation and classification problems. This framing aligns closely with challenges addressed in video action segmentation, where the goal is to identify and classify temporal boundaries of actions within a video. By adapting techniques from this domain, we aim to robustly detect room transitions and assign accurate room labels to each segment, leveraging the spatial and temporal cues inherent in walkthrough videos.

Our main task is to split up the input walkthrough video into clips, each containing only one room. The boundaries of each segment should ideally be at the moment the camera walks through the door opening connecting one room to the next, or crossing the imaginary line between different functions of an open-plan room. Additionally, each time segment should be labelled with the correct room type.

To the best of our abilities, we have not found previous work that was specifically targeted at real estate videos. However, the task of video action segmentation is very related. Here, an untrimmed video is segmented in separate time segments, each containing a distinct action of the filmed subject. The difference with our problem is that the video does not need to be segmented in terms of *where* it is captured (the room), but *what* the subject is doing (the action). Typical benchmark datasets contain actions like cooking a certain recipe (Kuehne et al., 2014; Fathi et al., 2011; Stein and McKenna, 2013) or toy or furniture assembly (Sener et al., ; Ben-Shabat et al., 2020; Ragusa et al., 2020).

Action segmentation aims to classify each frame of a video with a specific action label, akin to image segmentation, where each pixel is assigned a label. This frame-by-frame classification allows for a detailed temporal understanding of activities within a video.

The challenge of action segmentation lies in its need to handle varied action lengths, complex transitions between actions, and diverse video contexts. Unlike static image segmentation, action segmentation must account for temporal dependencies and dynamic variations, requiring sophisticated models that can learn and generalize from sequential data.

Current approaches split this task into extracting low-level spatial features and applying a high-level temporal classifier. There has been extensive work on the former. For the temporal aspect, a sliding window technique is applied in (Rohrbach et al., 2016; Ni et al., 2016). Building further upon the research of LSTM, (Gammulle et al., 2017) uses the convolutional layer outputs as input for an LSTM approach to detect human actions. Proving that the LSTM approach is suitable for the action segmentation task. Lea et al.(Lea et al., 2017b) use a convolutional encoder-decoder strategy with temporal convolutions to improve the extraction of temporal information, which outperforms the LSTM and is faster to train.

Transformer networks, which leverage a self-attention mechanism, have demonstrated remarkable effectiveness in processing sequential data (Vaswani et al., 2017). However, as noted by Yi et al. (Yi et al., 2021), applying Transformers to the action segmentation task presents several challenges. These include the absence of inductive biases, which be-

Figure 1: Example of a video tour sampled at 1 FPS (video viewable at https://youtu.be/SxVyLtyndCk).

comes particularly problematic when working with small datasets, difficulties in handling long input sequences due to the quadratic complexity of the self-attention mechanism, and limitations in the decoder architecture's ability to model temporal relations between multiple action segments, which is crucial for refining initial predictions. To address these issues, Yi et al. propose an encoder-decoder architecture, refining the output sequence through incremental decoding. Similarly, (Ji et al., 2022) introduces a multimodal Transformer, which uses a fusion of text and image data to perform the temporal video segmentation task.

Despite these advancements, challenges such as long sequence processing, inductive bias, and effective temporal modelling remain central to the action segmentation problem. Various approaches, including encoder-decoder architectures and multimodal Transformers, underscore the flexibility of these models, but there remains significant potential for improving the capture of long-range dependencies and refining segment predictions across diverse video contexts. In this work, we build upon these foundations by introducing a ResNet-Transformer approach, which integrates transition detection with the output sequence to enhance segmentation accuracy.

3 DATASET

The dataset used in this work is custom-made by the company that creates these interactive videos. It encompasses a diverse array of properties, including villas, houses, flats, offices, and student dorms. Specifically, the dataset includes 839 different properties,

Figure 2: Challenges in room type classification because of ambiguous labels. left: a living room and a bedroom. right: a bedroom and an attic.

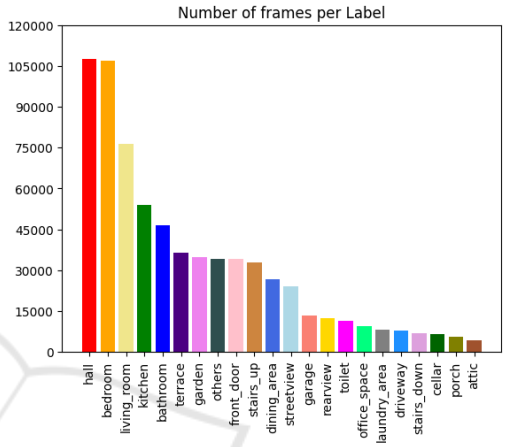

Figure 3: Histogram of room type labels in the dataset (room label colour codes are consistent through all figures in this paper).

each captured in multiple videos that have been labelled for the room classification task.

3.1 Room Classification Labels

For each property, the videos were manually labelled to identify various rooms. This process involved segmenting the videos and assigning appropriate room names to each segment. Unfortunately, as the problem is complex and sometimes ambiguous, we noted inconsistencies in the room labels. In Figure 2 we can see similar visual content representing different rooms. In total, we have a set of 22 room classes with a highly imbalanced distribution, as shown in Figure 3.

The *others* class contains all exotic classes like stables, swimming pool, technical room, treehouse, elevator, bicycle storage, library, etc. The classes are highly imbalanced, with nearly, 107500 hallway frames and only 4200 frames that show an attic. This labelled dataset supports the development and evaluation of our temporal segmentation model.

We split up the dataset using Stratified Random Sampling (May et al., 2010) to ensure a balanced representation of the dataset across training, validation,

and test sets. This approach ensures that the distribution of key attributes (e.g., class labels, video duration, number of different rooms, FPS, ...) is preserved in each split. By employing this technique, we avoid potential biases introduced by uneven class distributions and ensure that all models are evaluated on a representative sample of the data. For this study, we split the dataset into 70% real estate properties for training, 15% properties for validation, and 15% properties for testing. Figure 1 shows an example of an entire house sampled at 1FPS.

4 METHOD

In the following section, we will discuss the proposed methods for solving the smart video editing task. We will split the pipeline up in two distinct tasks: room classification (section 4) and transition detection (section 4)

To address the room type classification problem, we propose a ResNet-Transformer network. In our approach, a ResNet18 model serves as the frame-based spatial feature extractor, while a Transformer is used to capture temporal information and refine the output. Each frame in the video is processed by the ResNet18, which converts it into a 512-dimensional latent vector. These latent vectors, derived from a sequence of consecutive frames (*sequence length*), are then fed into a Transformer encoder to capture temporal dependencies.

The Transformer model consists of 5 encoder layers, each with 2 attention heads and a feed-forward dimension of 2048. Positional encodings are added to the input latent vectors to preserve the temporal order of the frames. The output from the Transformer encoder is passed through a classification head, which predicts the room type for each frame in the sequence.

The entire network is trained for 37 epochs end-to-end, using sequences of 23 frames and a batch size of 16. The model is optimized with a fixed learning rate of 1e-6, and data augmentation techniques are applied to the input frames to mitigate overfitting. The whole approach is shown in Fig 4.

To refine the outputs of the room classification network, we integrate doorway detections as a post-processing step. Doorframes act as distinctive visual cues, helping to clarify room boundaries and enhance classification accuracy. Our doorway detection system is built on a ResNet18 architecture for feature extraction, followed by a fully connected network that models temporal dependencies across a sliding window of multiple frames. Figure 5 gives an overview of the used technique.

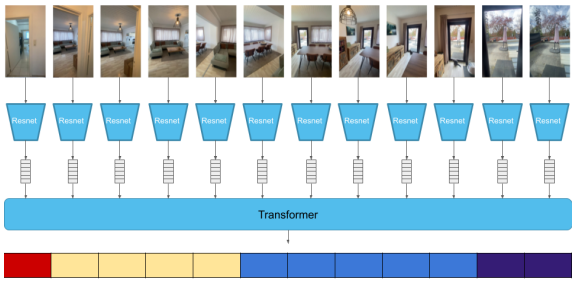

Figure 4: Architecture overview of the proposed network for room classification. Colour legend for room types: see Fig. 3.

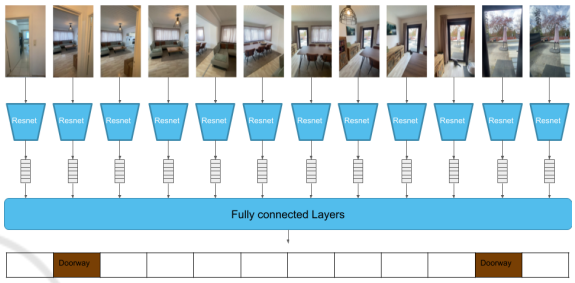

Figure 5: Architecture overview of the proposed network for doorway detection.

This network is trained on a subset of the original dataset, selecting transitions between “hall” and other room types, as these always involve a door. Figure 6 shows two samples of this subset. This approach ensures that only actual doorways are used, avoiding “open” transitions that could confuse the model. While this means we miss doorframes outside of hall transitions, this limitation is mitigated by the network’s ability to easily recognize door frames, allowing effective training even with a smaller dataset.

The network used for doorway detection employs a ResNet backbone to extract spatial features from individual frames, followed by three fully connected layers that aggregate information across multiple frames to capture temporal relationships. After initial experimentation, we empirically chose the window size for doorway detection to be eleven frames. An overview of this method is illustrated in Figure 5, a video example can be viewed here: https://youtu.be/2JpKkCI5dGc.

Once doorways are detected, their information is used in the post-processing stage to refine the

Figure 6: two samples that show a doorway transition.

segmentation borders predicted by the Transformer-based room classification network. We employ handcrafted rules to integrate the doorway detections into the room classification pipeline. The detection of a doorway signals the boundary between two different rooms, prompting an adjustment in the room classification output. Once we detect a doorway, we use a sliding window between two detected doorways. Additionally, we merge segments that are too small with their neighbouring segments respecting doorway positions, if applicable. With the help of these simple rules, we improve the predictions, aligning them more closely with the ground truth annotations.

As we will demonstrate in Section 5, this combination of deep learning-based doorway detection and rule-based post-processing enables more accurate segmentation of indoor environments by refining room classifications based on explicit structural cues.

5 RESULTS

5.1 Evaluation Metrics Used

In the temporal segmentation domain, several common metrics are used to assess the performance of models. We chose to report the F1 score, frame-wise accuracy, and the edit score, providing a comprehensive evaluation of both appearance-based and temporal prediction quality.

5.1.1 Frame-Wise Accuracy

Frame-wise accuracy measures the percentage of frames in the video that are correctly classified. While this metric provides a straightforward measure of classification performance, it can be misleading in temporal segmentation tasks. A model may achieve high frame-wise accuracy by correctly classifying the majority of frames, yet fail to capture the correct transitions between segments. Moreover, spikes and fast glitches are not penalized. Thus, while useful, frame-wise accuracy should be interpreted with caution when assessing temporal consistency.

5.1.2 F1 Score

The F1 score balances precision and recall by calculating the harmonic mean between the two. For temporal segmentation tasks, the F1 score is computed by comparing each predicted time segment with the ground truth through the Intersection over Union (IoU). A predicted segment is considered a true positive (TP) if its IoU with the corresponding ground truth segment exceeds a certain threshold. To capture the model's performance across different levels of strictness, we report F1 scores at three IoU thresholds: F1@0.10, F1@0.25, and F1@0.50.

However, one limitation of the F1 score is that it focuses on individual segments and does not capture the sequence-level structure of the predictions. It may overlook how well the overall segmentation aligns with the true sequence of events.

5.1.3 Edit Score

The edit score (Lu and Elhamifar, 2024) offers a complementary perspective by evaluating the sequence structure of predicted segments in relation to the ground truth. It measures the number of operations required to transform the predicted segmentation into the correct ground truth segmentation. Specifically, it uses the Levenshtein distance. The fewer operations required to correct the predicted segmentation, the higher the edit score. A key advantage of the edit score is its alignment with human post-processing effort. The operations counted by the edit score (insertion, deletion, replacement) directly correspond to the actions a human would need to take to manually correct the model's output.

By jointly observing the F1 score, frame-wise accuracy, and edit score, we obtain a more holistic view of model performance, balancing frame-level precision with the consistency and correctness of predicted segment sequences.

5.2 Model Architectures

Below, we describe the various model architectures tested out in our experiments, ranging from a baseline ResNet frame-by-frame model to more sophisticated temporal models utilizing fully connected layers, LSTMs, and transformers.

5.2.1 ResNet Frame-Based Classifier

As a baseline, we utilized a pre-trained ResNet-18 model to classify individual video frames (He et al., 2015). ResNet-18 was selected for its demonstrated ability to capture detailed appearance information, essential for distinguishing between different types of rooms. In this setup, each frame was treated as an independent entity without any form of temporal processing or sequential modeling. This approach allowed us to assess the model's performance based solely on appearance features, serving as a point of comparison for subsequent models incorporating temporal dynamics.

5.2.2 ResNet + Temporal Modeling via Fully Connected Network

In this variation, temporal relationships between frames were captured using a dense fully connected network. After extracting embeddings from each consecutive frame using the pretrained ResNet from sec. 5.2.1, the frame sequences were passed through three fully connected layers. This approach enables the model to aggregate spatial information across multiple frames, offering a richer and more dynamic representation of the scene. By processing frame sequences, the dense layers can capture short-term temporal dependencies and improve overall accuracy in scene classification tasks. We used a sequence length of 23 frames and two fully connected layers with a dimension of 1024 and 512 respectably.

5.2.3 ResNet + Temporal Modeling via LSTM

To capture more complex and long-term temporal dependencies, we employed a Long Short-Term Memory (LSTM) network. In this architecture, the ResNet-extracted embeddings from consecutive frames were fed into the LSTM, allowing the model to learn the temporal dynamics inherent in video sequences. This is particularly advantageous in understanding gradual transitions and movements between rooms, as the LSTM can retain information from earlier frames and use it to inform later predictions. The ability to model long-range temporal dependencies offers improved robustness, especially in scenarios where frame-by-frame spatial features alone may be insufficient to capture room transitions.

5.2.4 ResNet + Temporal Modeling via Transformer

To further enhance the modeling of temporal relationships, we experimented with a transformer-based architecture. Transformers have been shown to excel in sequence modeling tasks, primarily due to their self-attention mechanism, which can capture both short- and long-range dependencies. In this setup, the ResNet-extracted frame features were processed by transformer layers, allowing the model to attend to multiple frames simultaneously and to better capture context across a video sequence. Positional encoding was applied to preserve the sequential nature of the frames, and the model processed batches of 23 frames at a time. This setup provided a more context-aware interpretation of the video and significantly improved the understanding of the temporal structure of the scene.

5.3 Post-Processing Strategies

In addition to the different model architectures, we explored various post-processing techniques to further refine the temporal predictions made by the models. These strategies focus on improving robustness against frame-level misclassifications and ensuring that the model captures the transitions between different rooms more accurately. In table 1 the base value is without any post-processing.

5.3.1 Sliding Window with Majority Voting

In this approach, we introduced a sliding window method that applies majority voting across consecutive frames. For each segment of the video, the model's predictions over the sliding window are aggregated, and the most common prediction is assigned to the middle frame. This technique improves temporal consistency by smoothing out short-term misclassification spikes and ensuring that the final prediction is representative of the broader context. After experimentation, a window size of 8 frames was found to provide the best balance between smoothing and responsiveness to changes in the video sequence.

5.3.2 Transition Detection with Post-Processing Refinement

To further enhance the accuracy of transition placement, we employed the doorway detection network described in section 4 in conjunction with the room classification model. By combining the outputs of both the doorway detection and the sliding window based room classification models, we were able to apply rule-based logic to significantly improve performance. The post-processing step corrects the model's predictions by adjusting segment boundaries, ensuring that detected transitions correspond more accurately to actual changes in the environment, such as when moving between rooms. This integration of appearance and transition cues proves highly effective in reducing misclassifications during room changes and refining the overall segmentation logic, leading to more precise and robust scene interpretations.

5.4 Results Overview

Table 1 shows an overview of our temporal room segmentation and classification ablation study, where the final row demonstrates the superiority of our proposed method, scoring best on all evaluation metrics. We both compare four different model architectures (described in Section 5.2), each combined with three different post-processing steps used (handled in Section

Table 1: Evaluation of different models with various post-processing methods. Acc refers to frame-wise accuracy, while F1 scores are reported at three different IoU thresholds. The Edit score reflects the sequence-level correction needed for a perfect model output.

Model	Post-Processing Method	ID	Acc	F1@0.10	F1@0.25	F1@0.50	Edit
ResNet only	None	A	0.65	0.23	0.20	0.13	0.15
	Sliding Window	B	0.66	0.43	0.39	0.28	0.31
	S.W. + Transition Detection	C	0.66	0.51	0.47	0.35	0.38
ResNet + Fully Connected	None	D	0.61	0.25	0.23	0.17	0.20
	Sliding Window	E	0.61	0.48	0.45	0.34	0.39
	S.W. + Transition Detection	F	0.61	0.57	0.54	0.41	0.48
ResNet + LSTM	None	G	0.44	0.24	0.19	0.09	0.18
	Sliding Window	H	0.45	0.32	0.27	0.14	0.25
	S.W. + Transition Detection	I	0.46	0.36	0.30	0.16	0.27
ResNet + Transformer	None	J	0.65	0.38	0.35	0.26	0.29
	Sliding Window	K	0.66	0.57	0.52	0.40	0.47
	S.W. + Transition Detection	L	**0.66**	**0.61**	**0.56**	**0.43**	**0.51**

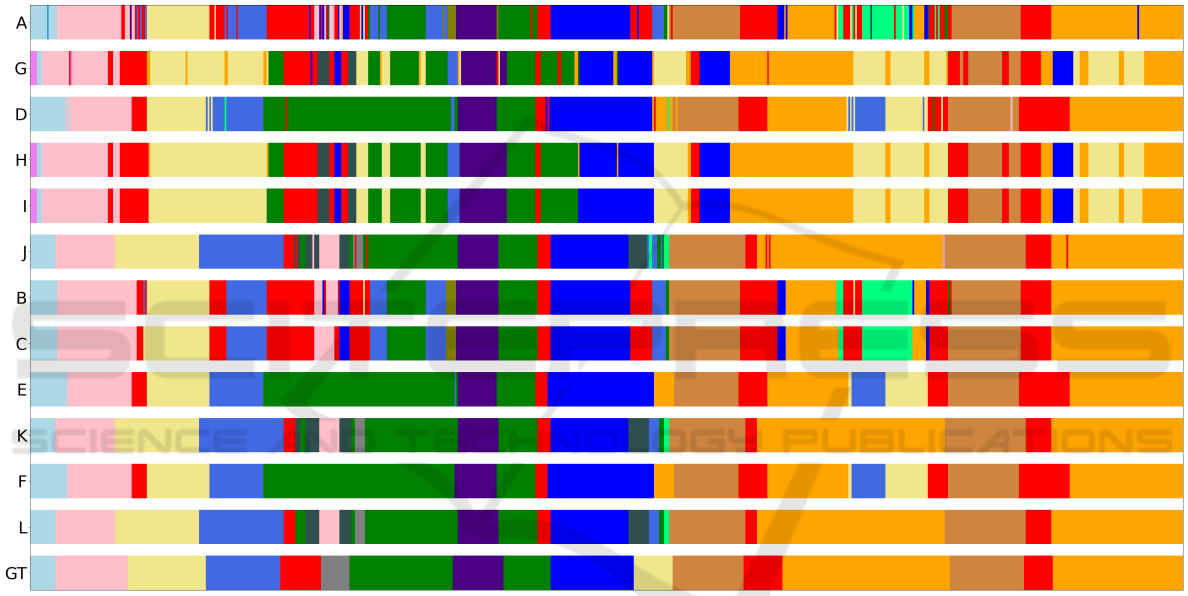

Figure 7: Resulting room segmentation timelines of the different model and post-processing method combinations from table 1 on a test video. From top to bottom: pipeline IDs ordered in increasing *Edit Score* order. GT: Ground Truth. (Input video viewable at https://youtu.be/SxVyLtyndCk?si=Hz3ZEoOw6HhqQrGt. Colour legend for room types: see Fig. 3).

5.3) to refine the model output. Both model and post-processing method increase in complexity in towards the bottom of the table. The quantitative evaluation measures used are defined in section 5.1.

As a qualitative result, Figure 7 shows the output of each of the models in this ablation study on a video. As can be observed, our final pipeline (pipeline ID "L") consisting of a ResNet frame-based embedding extractor, Transformer-based temporal modelling and a refinement stage using our custom trained doorway detector matches best with the ground truth room label sequence. An second output example, with indication of room labels (ground truth in red, predictions in blue) can be viewed here: https://youtu.be/KBeduh7AjaA.

All our models are trained on the dataset described in Section 3. Each video was annotated with the corresponding room types at each frame, forming the ground truth labels necessary for training and evaluation. Each frame of a video was resized to 224×224 pixels, we also performed data augmentation techniques, such as colour jitter, random cropping, etc., to enhance the model's robustness. All the videos were subsampled from 60FPS to 12FPS in order to view a larger timeframe and prevent overfitting.

6 CONCLUSION

This work presents an effective automated system for creating interactive real estate video tours by addressing room classification and transition detection. The ResNet-Transformer network demonstrated strong capabilities in capturing both spatial and temporal features for accurate room classification. The Transformer-based model improved more than 20% as compared to a more traditional LSTM sequence processing. The integration of door transition detection as a post-processing step enhanced the performance across all models. Indeed, detected door transitions contribute essential structural information, particularly aiding in the accurate delineation of room boundaries when doors were present. This approach improved the overall precision and ensured more consistent room layout predictions, paving the way for more sophisticated applications in automated video editing systems, particularly in real estate domain. In future work, we plan to do user satisfaction studies.

ACKNOWLEDGEMENTS

This work has partially funded by the VLAIO project WAIVE and the real estate video company.

REFERENCES

Ben-Shabat, Y., Yu, X., Saleh, F., Campbell, D., Rodriguez-Opazo, C., Li, H., and Gould, S. (2020). The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose.

Fathi, A., Ren, X., and Rehg, J. M. (2011). Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288.

Gammulle, H., Denman, S., Sridharan, S., and Fookes, C. (2017). Two Stream LSTM: A Deep Fusion Framework for Human Action Recognition. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 177–186.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Ji, L., Wu, C., Zhou, D., Yan, K., Cui, E., Chen, X., and Duan, N. (2022). Learning temporal video procedure segmentation from an automatically collected large dataset. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2733–2742.

Kuehne, H., Arslan, A. B., and Serre, T. (2014). The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*.

Lautz, J., Snowden, B., and Dunn, M. (2014). Chief economist and senior vice president.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2016). Temporal convolutional networks for action segmentation and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (arXiv:1611.05267). arXiv:1611.05267 [cs].

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017a). Temporal convolutional networks for action segmentation and detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012, Los Alamitos, CA, USA. IEEE Computer Society.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017b). Temporal Convolutional Networks for Action Segmentation and Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012, Honolulu, HI. IEEE.

Lu, Z. and Elhamifar, E. (2024). FACT: Frame-action cross-attention temporal modeling for efficient supervised action segmentation. In *Conference on Computer Vision and Pattern Recognition 2024*.

May, R., Maier, H., and Dandy, G. (2010). Data splitting for artificial neural networks using som-based stratified sampling. *Neural Networks*, 23(2):283–294.

Miech, A., Alayrac, J.-B., Smaira, L., Laptev, I., Sivic, J., and Zisserman, A. (2020). End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*.

Ni, B., Yang, X., and Gao, S. (2016). Progressively Parsing Interactional Objects for Fine Grained Action Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1020–1028, Las Vegas, NV, USA. IEEE.

Ragusa, F., Furnari, A., Livatino, S., and Farinella, G. M. (2020). The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain.

Rohrbach, M., Rohrbach, A., Regneri, M., Amin, S., Andriluka, M., Pinkal, M., and Schiele, B. (2016). Recognizing Fine-Grained and Composite Activities using Hand-Centric Features and Script Data. *International Journal of Computer Vision*, 119(3):346–373. arXiv:1502.06648 [cs].

Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., and Yao, A. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. *CVPR 2022*.

Stein, S. and McKenna, S. J. (2013). Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, page 729–738, New York, NY, USA. Association for Computing Machinery.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Yi, F., Wen, H., and Jiang, T. (2021). Asformer: Transformer for action segmentation.